Application Note:

How to generate your keys, flash to xPico240/250, build the SDK code and sign the key for the firmware

This app note applies to the following Lantronix Products:

xPico 240
xPico 250

Lantronix, Inc.
15353 Barranca Parkway
Irvine, CA 92618
Tel: +1 (949) 453-3990
Part Number 920-???
Revision A June 2008

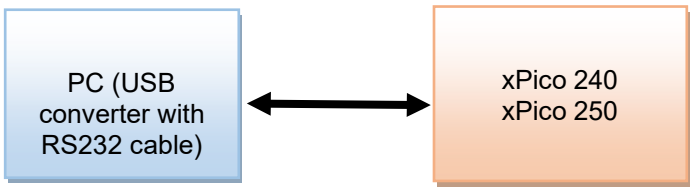
Overview

This application note provides more guidance in loading the firmware with xPico 240/250 secure boot enable.

Initial setup

For the following instructions, connect the serial line between a xPico 200 series Evaluation Board and a PC.

Prior to running the example, download and install Tera Term and Cygwin to open serial line and network connections.



Steps:

1. Generate key by using the following two commands. First, you generate the private key. Then, you use the private key to generate the public key. If you are using windows system, you can install Cygwin which is Linux base terminal supporting "openssl" commands. To know where you keys locate, you can set the path in Cygwin to your C:/xPico200/lantronix/bin
    - openssl genrsa -f4 -out oem_rsa_key.priv 2048
    - openssl rsa -in oem_rsa_key.priv -pubout -out optional_rsa_key.pub

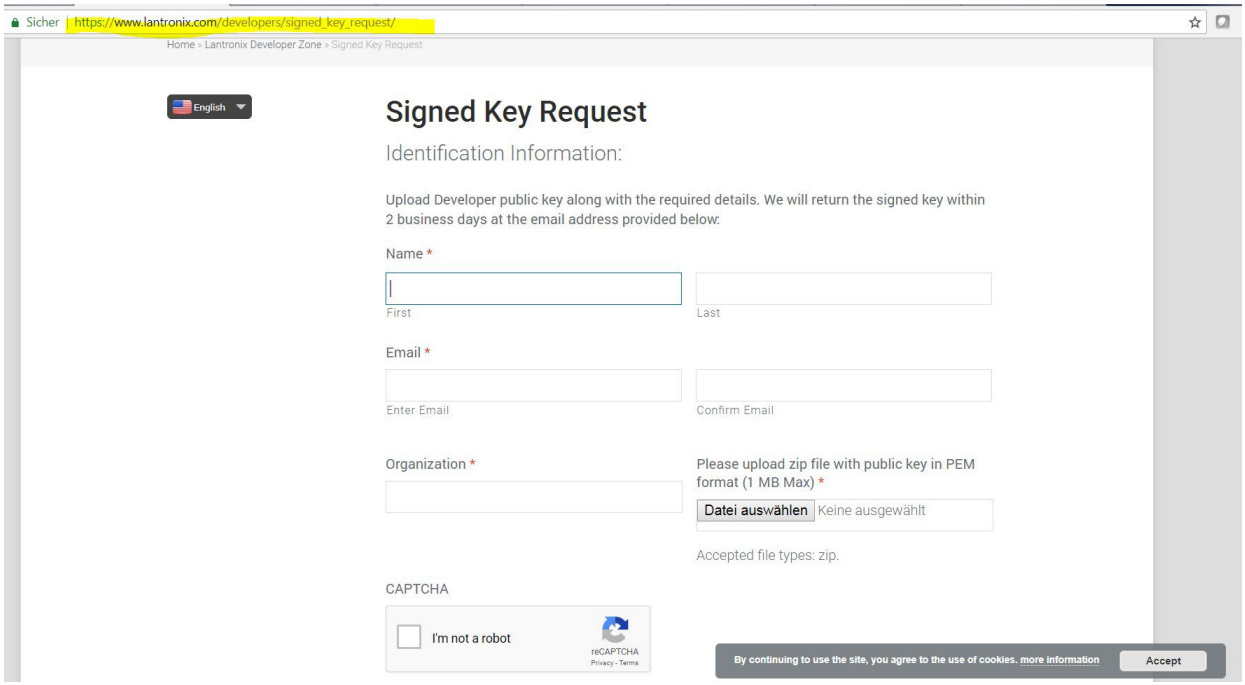
```
blee@Brooke-PC /cygdrive/c/xPico200
$ openssl genrsa -f4 -out oem_rsa_key.priv 2048
Generating RSA private key, 2048 bit long modulus
.....................................................+++
..+++
e is 65537 (0x10001)
```

```
blee@Brooke-PC /cygdrive/c/xPico200
$ openssl rsa -in oem_rsa_key.priv -pubout -out optional_rsa_key.pub
writing RSA key
```

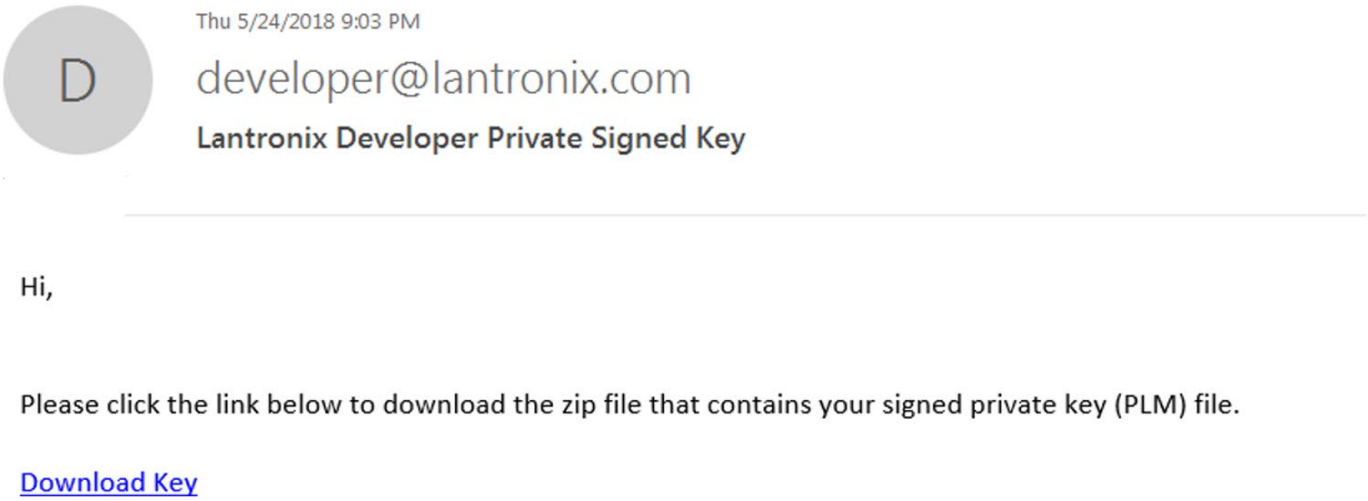
2. Please zip the “optional_rsa_key.pub” and then submit it online via “Request Form”.

Name	Änderungsdatum	Typ	Größe
build	5/9/2018 4:28 PM	Dateiordner	
custom	5/9/2018 4:28 PM	Dateiordner	
documentation	5/9/2018 4:28 PM	Dateiordner	
env	5/9/2018 4:28 PM	Dateiordner	
html	5/9/2018 4:28 PM	Dateiordner	
Key	5/24/2018 10:24 PM	Dateiordner	
lantronix	5/9/2018 4:28 PM	Dateiordner	
MinGW	5/9/2018 4:29 PM	Dateiordner	
rules	5/9/2018 4:28 PM	Dateiordner	
toolchain	5/9/2018 4:28 PM	Dateiordner	
work	5/22/2018 10:47 A...	Dateiordner	
MinGW-shell	4/27/2018 8:37 PM	Verknüpfung	2 KB
oem_rsa_key.priv	5/24/2018 10:24 PM	PRIV-Datei	2 KB
optional_rsa_key	5/24/2018 10:28 PM	Microsoft Publishe...	1 KB
optional_rsa_key	5/24/2018 10:30 PM	WinZip File	1 KB

3. Send your public key (.pub) in zip format. Then, fill out the "Request Form" online with this zip file. After some time, you will receive the following email. You will download the zip file and then unzip it. Afterwards, you will use it in step 4 to flash it in your xPico240.



You will receive the following email.



4. After you download the key (zip file), please unzip it and then you will use "optional_rsa_key.pub.signed.rom" later to flash the key in your xPico 240.

Name	Änderungsdatum	Typ	Größe
optional_rsa_key.pub.signed.rom	5/22/2018 1:12 PM	ROM-Datei	1 KB
optional_rsa_key.pub.signed.sha1	5/22/2018 1:12 PM	SHA1-Datei	1 KB

5. Go to "C:\xPico200\" and double-click "MinGW-shell".

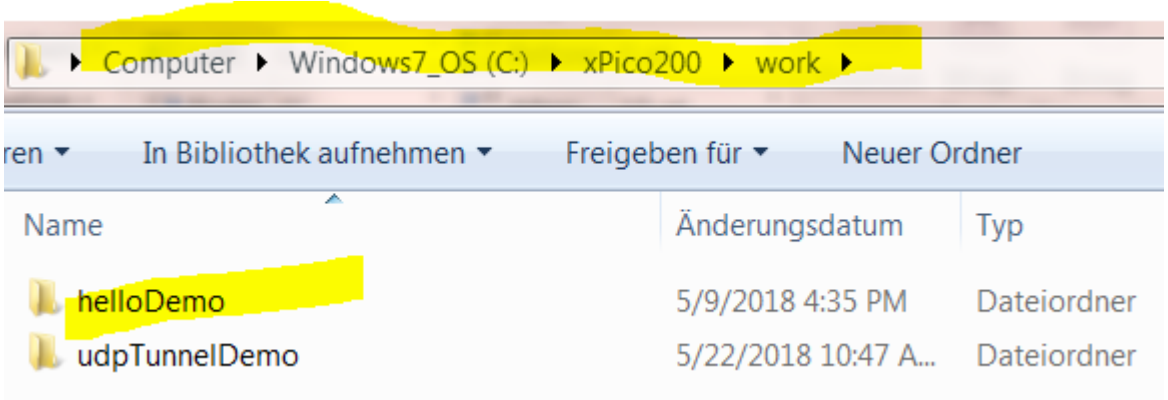
Name	Änderungsdatum	Typ	Größe
build	5/9/2018 4:28 PM	Dateiordner	
custom	5/9/2018 4:28 PM	Dateiordner	
documentation	5/9/2018 4:28 PM	Dateiordner	
env	5/9/2018 4:28 PM	Dateiordner	
html	5/9/2018 4:28 PM	Dateiordner	
Key	5/24/2018 10:35 PM	Dateiordner	
lantronix	5/9/2018 4:28 PM	Dateiordner	
MinGW	5/9/2018 4:29 PM	Dateiordner	
rules	5/9/2018 4:28 PM	Dateiordner	
toolchain	5/9/2018 4:28 PM	Dateiordner	
work	5/22/2018 10:47 A...	Dateiordner	
MinGW-shell	4/27/2018 8:37 PM	Verknüpfung	2 KB

```
MINGW32:/c/xPico200/custom/project/helloDemo

blee@Brooke-PC ~
$ cd c:/xPico200

blee@Brooke-PC /c/xPico200
$ cd custom/project

blee@Brooke-PC /c/xPico200/custom/project
$ ls
btOemProvisioningDemo  mach10SerialAPIDemo   tcpTunnelDemo
configurablePinDemo    powerDownDemo         udpTunnelDemo
customDataDemo         programmaticScanDemo  udpTunnelDemo_old
echoDemo               secureTunnelDemo      xmlAccessDemo
helloDemo              spiLogDemo

blee@Brooke-PC /c/xPico200/custom/project
$ cd helloDemo

blee@Brooke-PC /c/xPico200/custom/project/helloDemo
$ make
```

```
blee@Brooke-PC /c/xPico200/custom/project/helloDemo
$ make
udp_tunnel_old_module_defs.h
udp_tunnel_old_module_definitions.h
udp_tunnel_old_module_libs.h
all_of_the_module_definitions.h
work__helloDemo__modules__xml_access__xml_access_module_libs.dep
work__helloDemo__modules__xml_access__embedded_files.dep
work__helloDemo__modules__xml_access__module_datetime.dep
work__helloDemo__modules__xml_access__module_defs.dep
custom__module__xml_access__xml_access.dep
udp_tunnel_old_module_libs.c
work__helloDemo__modules__udp_tunnel_old__udp_tunnel_old_module_libs.dep
/c/xPico200/work/helloDemo/modules/udp_tunnel_old/embedded_files
work__helloDemo__modules__udp_tunnel_old__embedded_files.dep
module_defs.c
/c/xPico200/work/helloDemo/modules/udp_tunnel_old/module_datetime.c
work__helloDemo__modules__udp_tunnel_old__module_datetime.dep
work__helloDemo__modules__udp_tunnel_old__module_defs.dep
custom__module__udp_tunnel_old__udp_tunnel.dep
work__helloDemo__modules__udp_tunnel__udp_tunnel_module_libs.dep
work__helloDemo__modules__udp_tunnel__embedded_files.dep
work__helloDemo__modules__udp_tunnel__module_datetime.dep
work__helloDemo__modules__udp_tunnel__module_defs.dep
custom__module__udp_tunnel__wbcp.dep
custom__module__udp_tunnel__udp_tunnel.dep
work__helloDemo__modules__tcp_tunnel__tcp_tunnel_module_libs.dep
work__helloDemo__modules__tcp_tunnel__embedded_files.dep
work__helloDemo__modules__tcp_tunnel__module_datetime.dep
work__helloDemo__modules__tcp_tunnel__module_defs.dep
custom__module__tcp_tunnel__tcp_tunnel.dep
work__helloDemo__modules__spi_log__spi_log_module_libs.dep
work__helloDemo__modules__spi_log__embedded_files.dep
work__helloDemo__modules__spi_log__module_datetime.dep
work__helloDemo__modules__spi_log__module_defs.dep
custom__module__spi_log__spi_log.dep
work__helloDemo__modules__secure_tunnel__secure_tunnel_module_libs.dep
work__helloDemo__modules__secure_tunnel__embedded_files.dep
work__helloDemo__modules__secure_tunnel__module_datetime.dep
work__helloDemo__modules__secure_tunnel__module_defs.dep
custom__module__secure_tunnel__secure_tunnel.dep
work__helloDemo__modules__programmatic_scan__programmatic_scan_module_libs.dep
work__helloDemo__modules__programmatic_scan__embedded_files.dep
work__helloDemo__modules__programmatic_scan__module_datetime.dep
work__helloDemo__modules__programmatic_scan__module_defs.dep
custom__module__programmatic_scan__programmatic_scan.dep
```

```
MINGW32:/c/xPico200/custom/project/helloDemo
hello_world_partition1.elf
hello_world_partition1.elf
hello_world_partition1.stripped.elf
hello_world_partition1.bin
usb_host_partition1.elf
usb_host_partition1.stripped.elf
usb_host_partition1.bin
project_partition1.bin
project_partition1.rom
helloDemo_1.7.0.2R1.1.0_partition1.rom
hello_world_partition2.symbols.elf
hello_world_partition2.elf
hello_world_partition2.elf
hello_world_partition2.stripped.elf
hello_world_partition2.bin
usb_host_partition2.elf
usb_host_partition2.stripped.elf
usb_host_partition2.bin
project_partition2.bin
project_partition2.rom
helloDemo_1.7.0.2R1.1.0_partition2.rom
helloDemo_1.7.0.2R1.1.0.rom

Image Summary (Partition size is 0x1E0000):
  Flash Usage: 1211784 of 1966080 (61%, 754296 remaining)
  SRAM Usage:  636192 of 2097152 (30%, 1460960 remaining)

blee@Brooke-PC /c/xPico200/custom/project/helloDemo
$
```

6. Go to “C:\xPico200\work\ “. You will find the project name you just built.



7. Please copy the “.rom” file you built into “C:\xPico200\lantronix\bin\” folder which we will use it later to build the signed firmware.

dependencies	5/24/2018 10:52 PM	Dateiordner	
includes	5/24/2018 10:52 PM	Dateiordner	
lantronix	5/9/2018 4:35 PM	Dateiordner	
libraries	5/9/2018 4:35 PM	Dateiordner	
modules	5/24/2018 10:52 PM	Dateiordner	
objects	5/24/2018 10:52 PM	Dateiordner	
project	5/24/2018 10:52 PM	Dateiordner	
helloDemo_1.7.0.2R1.1.0.rom	5/24/2018 10:52 PM	ROM-Datei	2,367 KB
helloDemo_1.7.0.2R1.1.0_partition1.rom	5/24/2018 10:52 PM	ROM-Datei	1,184 KB
helloDemo_1.7.0.2R1.1.0_partition2.rom	5/24/2018 10:52 PM	ROM-Datei	1,184 KB

8. Go to “C:\xPico200\lantronix\bin” folder. We will use “ltrx-signimage.exe” later to build signed firmware.

Name	Änderungsdatum	Typ	Größe
helloDemo_1.7.0.2R1.1.0.rom	5/24/2018 10:52 PM	ROM-Datei	2,367 KB
helloDemo_1.7.0.2R1.signed.rom	5/24/2018 10:59 PM	ROM-Datei	2,368 KB
ltrx-manufacturing-test-loader.rom	4/27/2018 9:53 PM	ROM-Datei	706 KB
ltrx-mkimage	4/27/2018 9:53 PM	Anwendung	173 KB
ltrx-signimage	4/27/2018 9:53 PM	Anwendung	262 KB
oem_rsa_key.priv	5/22/2018 11:23 A...	PRIV-Datei	2 KB
oem_sdk_1.7.0.2R1.signed.rom	5/22/2018 7:46 PM	ROM-Datei	2,456 KB
optional_rsa_key	5/22/2018 11:24 A...	Microsoft Publishe...	1 KB
optional_rsa_key	5/24/2018 4:53 PM	WinZip File	1 KB
send_loader_115K.ttl	4/27/2018 9:53 PM	TTL-Datei	1 KB
udpTunnelDemo_1.7.0.2R1.1.0.rom	5/22/2018 10:47 A...	ROM-Datei	2,455 KB

- Use MinGW terminal to access C:/xPico200/lantronix/bin/ folder. Then type the following command.
  “ltrx-signimage.exe oem_rsa_key.priv oem_sdk_1.0.0.0R1.rom oem_sdk_1.0.0.0R1.signed.rom”

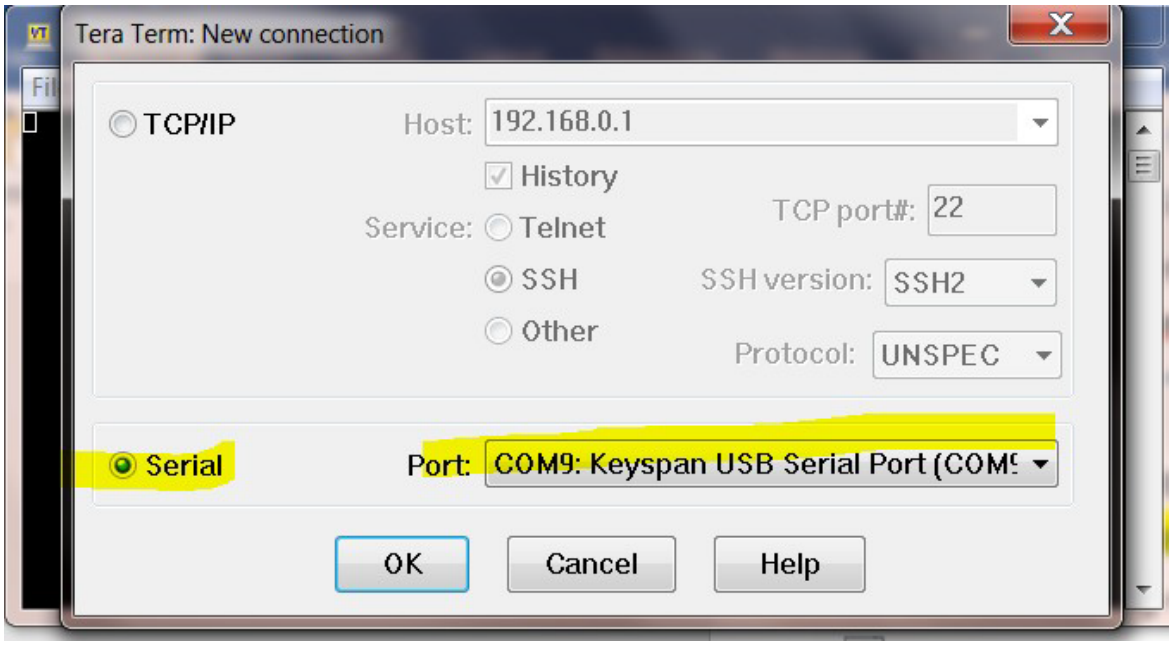
```
blee@Brooke-PC /c/xPico200/lantronix/bin
$ ls
helloDemo_1.7.0.2R1.1.0.rom          oem_sdk_1.7.0.2R1.signed.rom
ltrx-manufacturing-test-loader.rom   optional_rsa_key.pub
ltrx-mkimage.exe                     optional_rsa_key.zip
ltrx-signimage.exe                   send_loader_115K.ttl
oem_rsa_key.priv                     udpTunnelDemo_1.7.0.2R1.1.0.rom

blee@Brooke-PC /c/xPico200/lantronix/bin
$ ltrx-signimage.exe oem_rsa_key.priv helloDemo_1.7.0.2R1.1.0.rom helloDemo_1.7
.0.2R1.signed.rom
```

9. You will find the signed.rom firmware in the folder. This is the signed firmware with the key and SDK rom file.

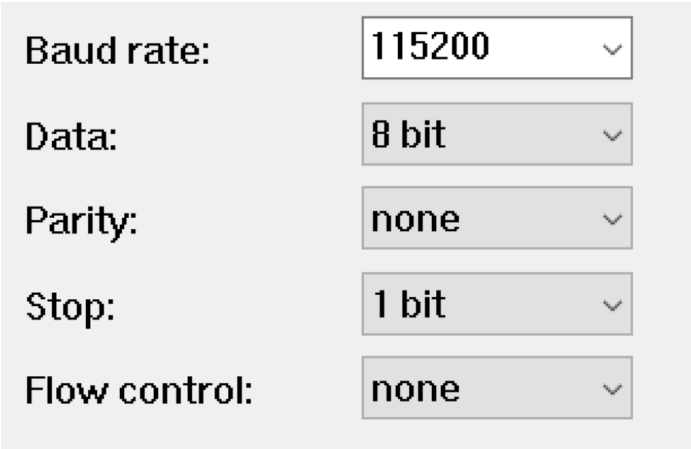
Name	Änderungsdatum	Typ	Größe
helloDemo_1.7.0.2R1.1.0.rom	5/24/2018 10:52 PM	ROM-Datei	2,367 KB
helloDemo_1.7.0.2R1.signed.rom	5/24/2018 10:59 PM	ROM-Datei	2,368 KB
ltrx-manufacturing-test-loader.rom	4/27/2018 9:53 PM	ROM-Datei	706 KB
ltrx-mkimage	4/27/2018 9:53 PM	Anwendung	173 KB
ltrx-signimage	4/27/2018 9:53 PM	Anwendung	262 KB
oem_rsa_key.priv	5/22/2018 11:23 A...	PRIV-Datei	2 KB
oem_sdk_1.7.0.2R1.signed.rom	5/22/2018 7:46 PM	ROM-Datei	2,456 KB
optional_rsa_key	5/22/2018 11:24 A...	Microsoft Publishe...	1 KB
optional_rsa_key	5/24/2018 4:53 PM	WinZip File	1 KB
send_loader_115K.ttl	4/27/2018 9:53 PM	TTL-Datei	1 KB
udpTunnelDemo_1.7.0.2R1.1.0.rom	5/22/2018 10:47 A...	ROM-Datei	2,455 KB

10. Connect xPico 240 with Serial cable (RS232 and USB converter) to your PC.

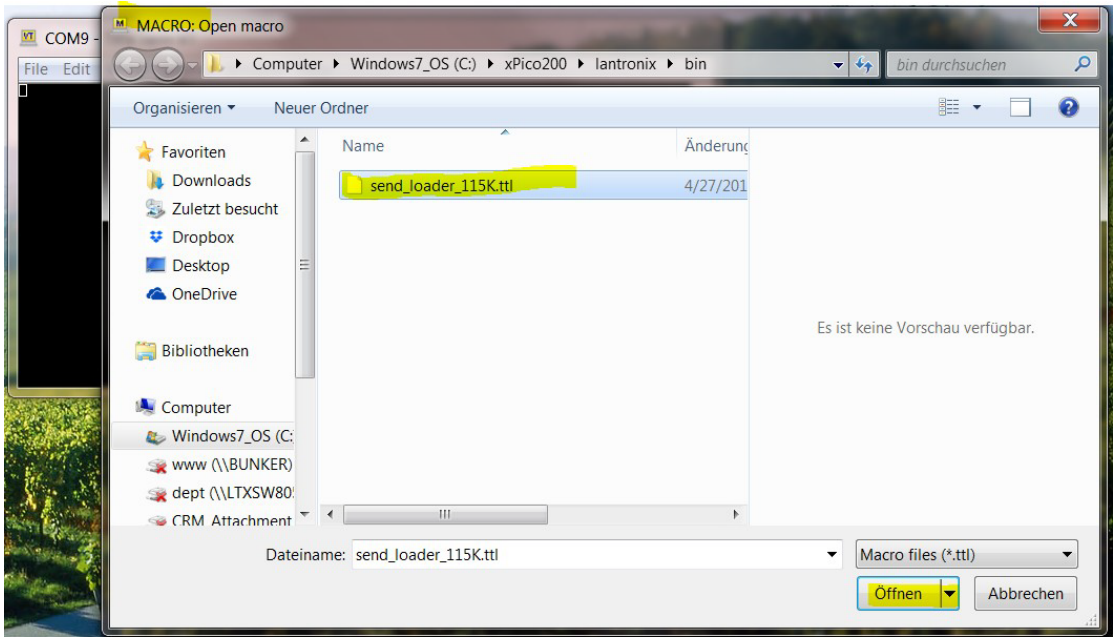
11. Install Tera Term which we will use later to run the script. Open Tera Term and serial port.



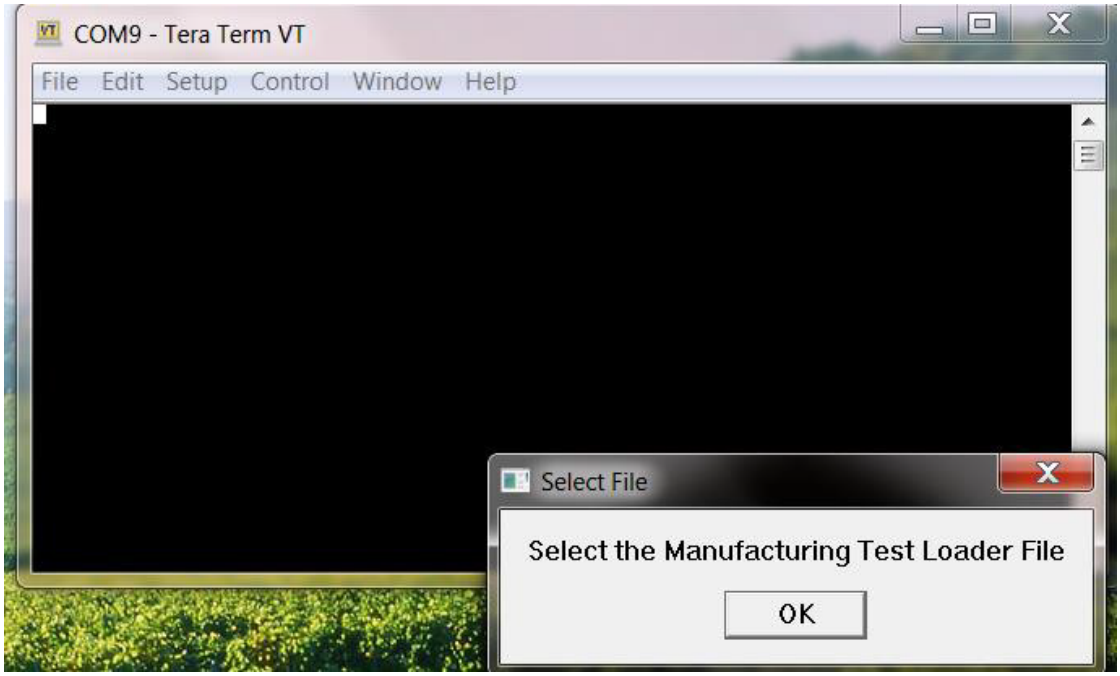
12. Setup the serial port

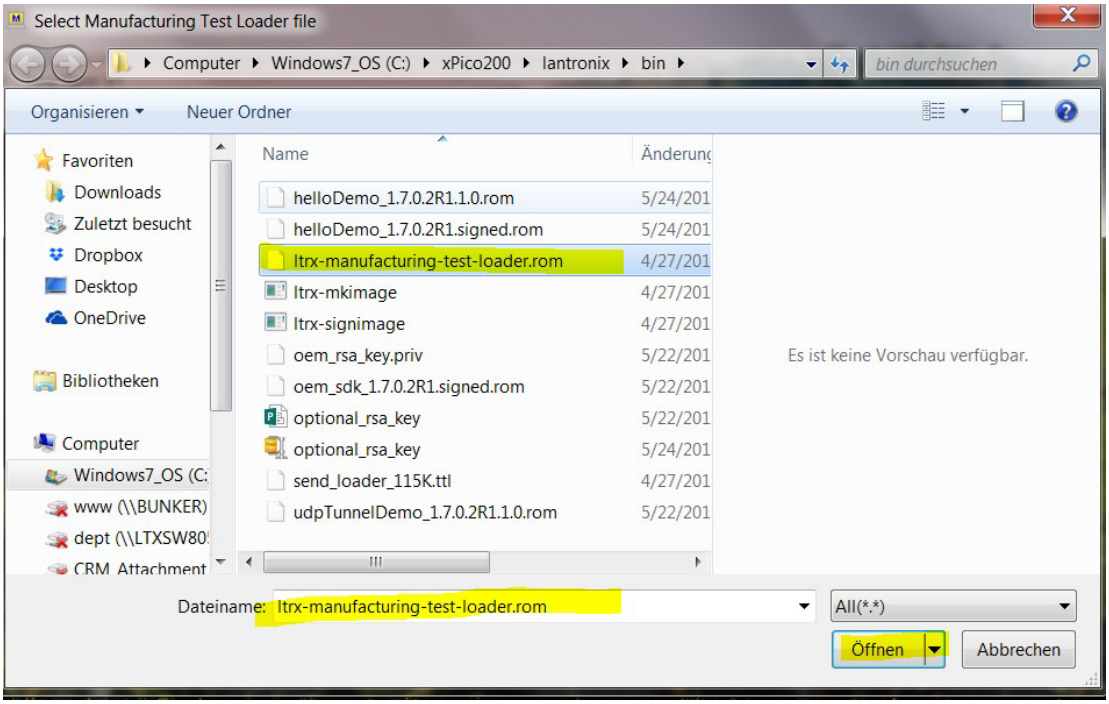


13. Click “Control -> Macro” and then select “C:/xPico200/lantronix/bin/send_loader_115K.ttl” or another location where it is stored.



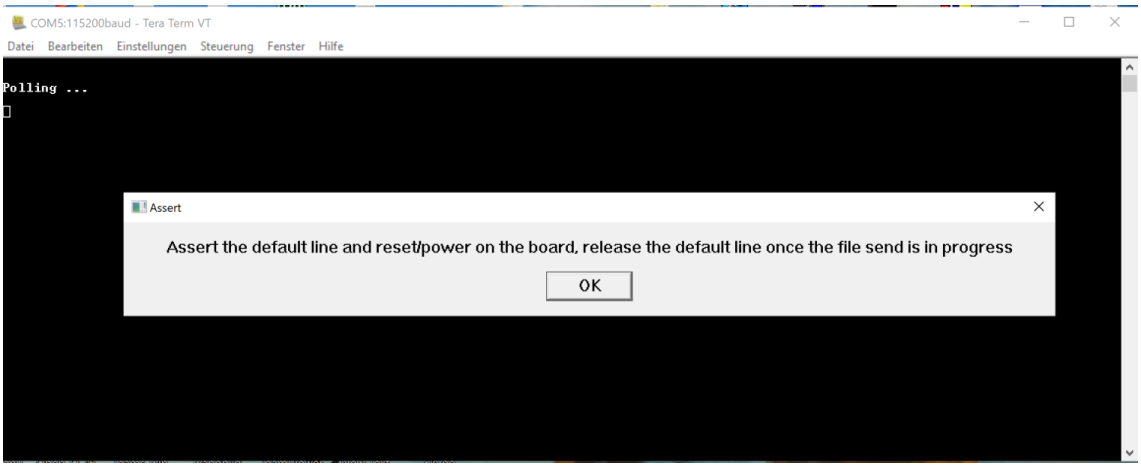
14. Select “C:/xPico200/lantronix/bin/ltrx-manufacturing-test-loader.rom” or another location where it is stored.



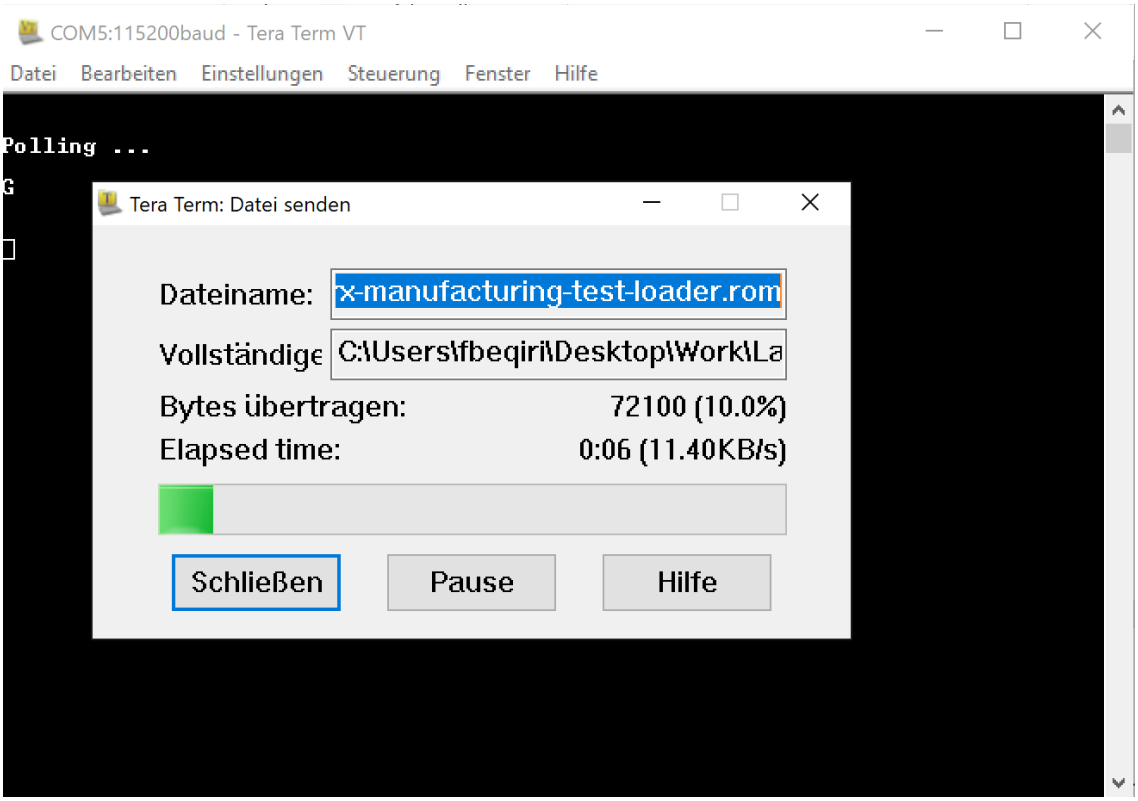


15. Click OK at the loader prompt.

With the power to the xPico 200 module turned off, hold down the **Defaults** button. Turn on the xPico 200 module. Release the **Defaults** button.

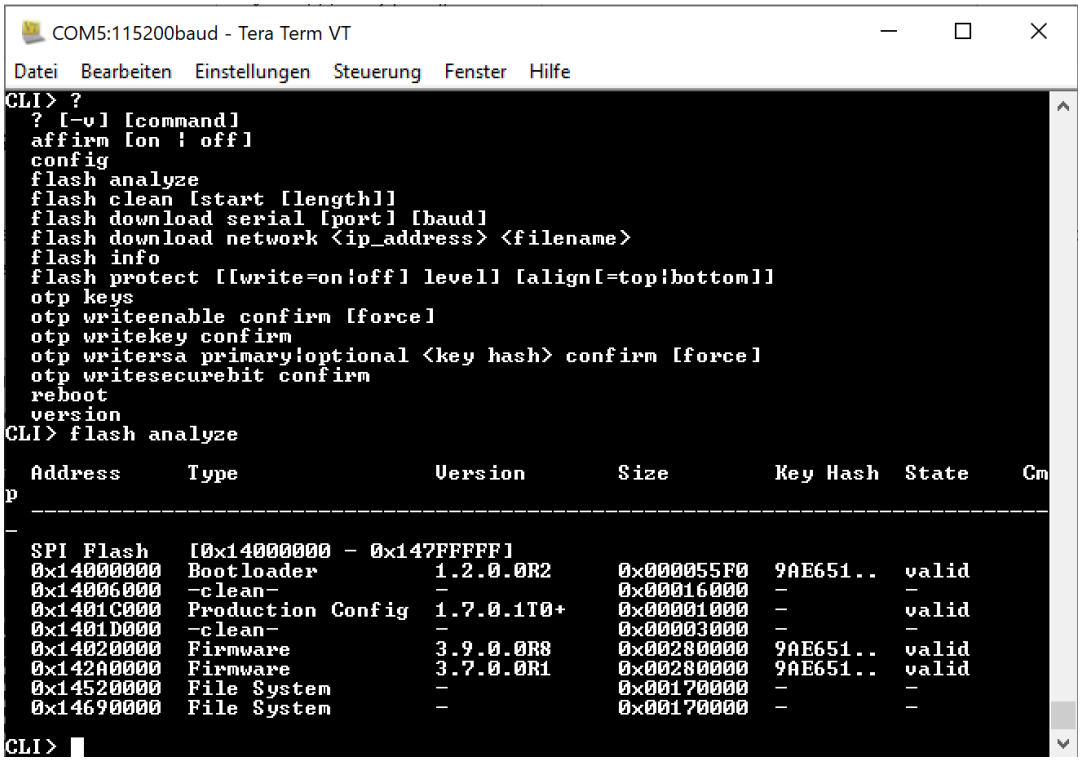


This will start the loader script sending the '!SL' sequence. The bootloader will examine the serial port, and finding the incoming '!SL', will respond with a 'G'. Once it receives the 'G', the script will send over the selected loader file.

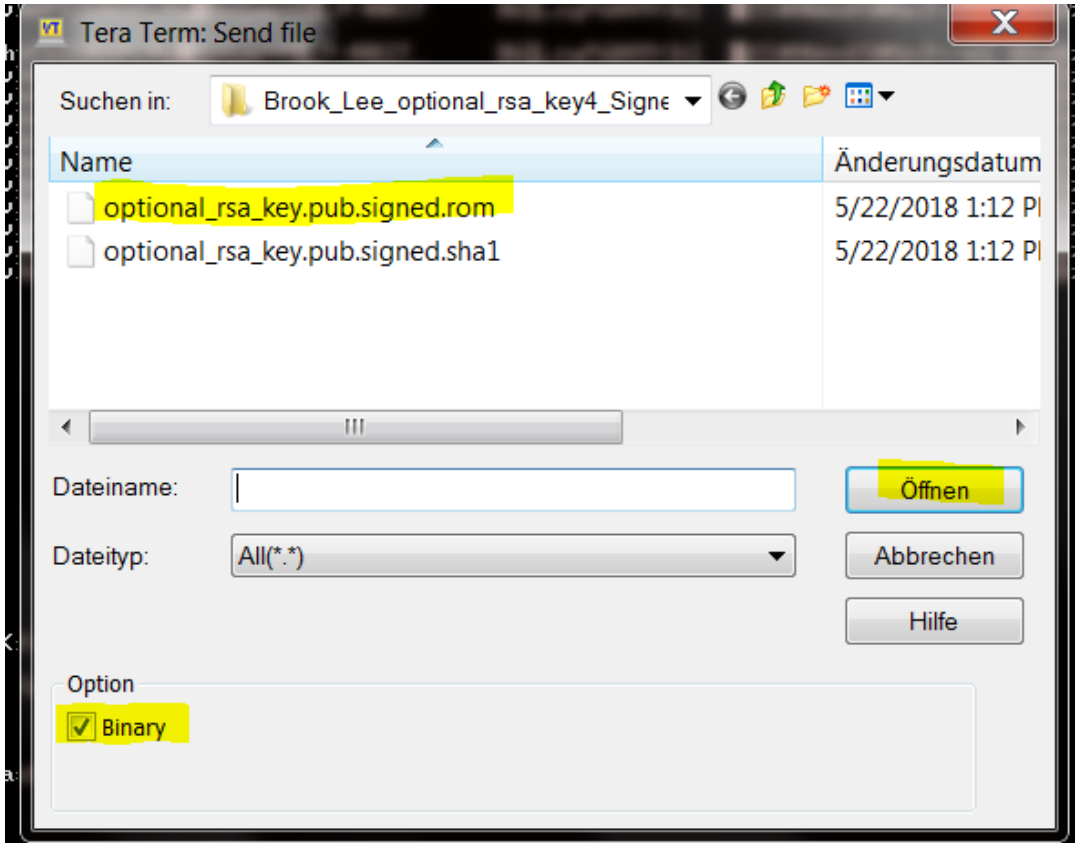


16. Then, you will enter CLI.
    This is an optional step, but can help. Type ? to have a look of commands. You can see the currently installed version of firmware using ‘flash analyze’, if there is one, and if it is valid. You can re-run this command after installing new firmware as a double check to ensure it’s installed correctly.

    Below is a capture of my xPico200. Yours may look a little different.



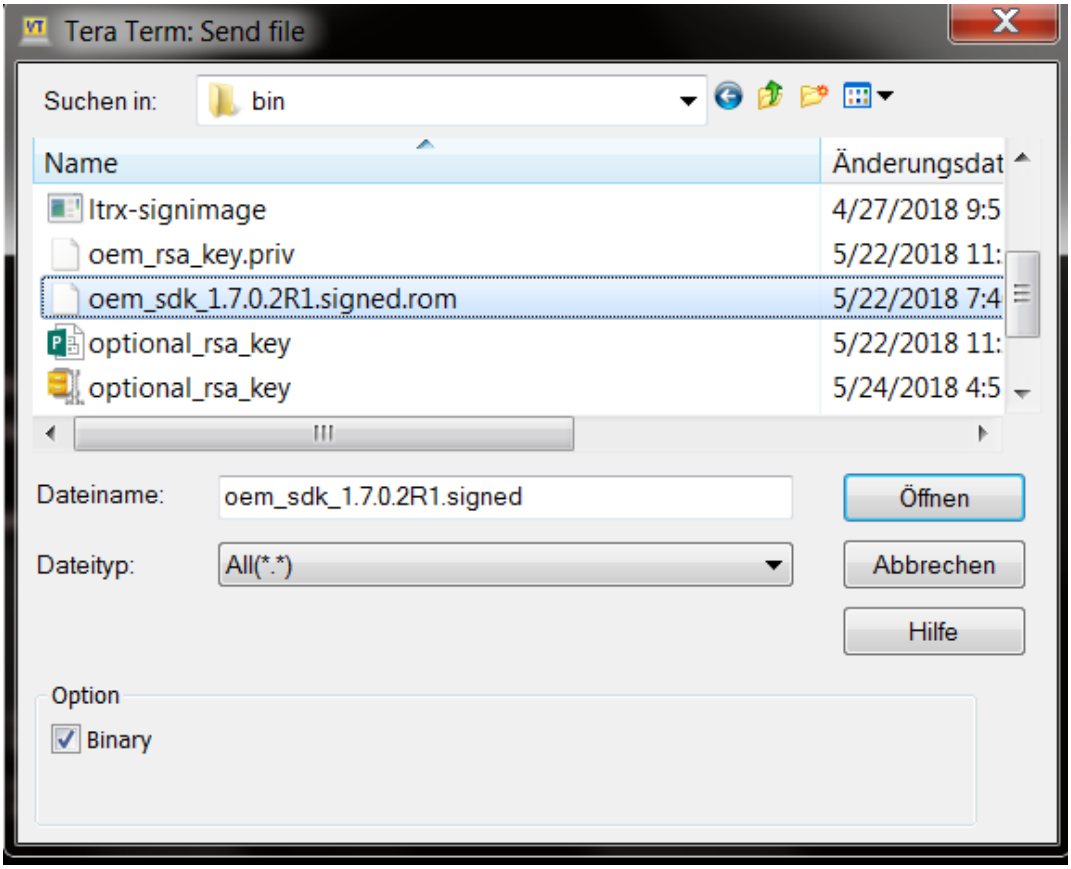
17. Enter “flash download serial” and then click “File -> Send File”, and select the rom file you unzip from Request Form -> click the check box of binary”.



18. You can enter “otp keys” to verify otp keys which you just wrote into the flash.

```
CLI> otp keys

OTP Keys and Status:

Primary RSA Key Hash (SHA256):
180231C4  9A E6 51 08 CB 50 7A 77 F4 DC C5 11 EB F8 DA F3  ..Q..Pzw........
180231D4  B9 0F 0D B2 F6 E9 CD AD D3 F2 D0 07 50 50 75 E2  ............PPu.
Primary RSA Key Hash Redundant:
180231A4  9A E6 51 08 CB 50 7A 77 F4 DC C5 11 EB F8 DA F3  ..Q..Pzw........
180231B4  B9 0F 0D B2 F6 E9 CD AD D3 F2 D0 07 50 50 75 E2  ............PPu.

Optional RSA Key Hash (SHA256):
18023100  CB 93 36 57 80 1B 9B 90 A2 D1 FB F0 B2 1D F7 2C  ..6W............,
18023110  26 B0 9C 18 A7 FE 77 C7 78 67 44 CC 7E AB 68 44  &.....w.xgD.~.hD

Secure Data Key: Configured

Boot Options:
18023248  19                                               .
Boot Options Redundant:
18023244  19                                               .

Secure Bit: Disabled

CLI>
```

19. You can enter “flash download serial” to flash the signed firmware.



When the file is first sent over, it stalls for a bit while the location in flash is erased.

After a minute or so, bytes will start flowing.

The firmware is successfully flashed.

```
CLI> flash download serial
Ensure that hardware flow control is enabled and send image now.
Installing:          Firmware
  Version:           1.7.0.2R1.1
  Product Code:      Y2
  Partition Size:    0x001E0000 (1966080)
  Image Size:        0x00132FD0 (1257424)
  Install Address:   0x14020000
Installing image to 0x14020000

Download finished, 1257424 total bytes
ERROR: Image in SPI flash failed contents verification
CLI>
```

Please enter “reboot”

```
CLI> Reboot
Rebooting ...
```

Conclusion:

This application note shows how to generate the private key and public key yourself, build the SDK code into rom file, build the firmware with the key to have signed firmware, load the key into xPico 240/250 memory and download signed firmware into xPico 240.